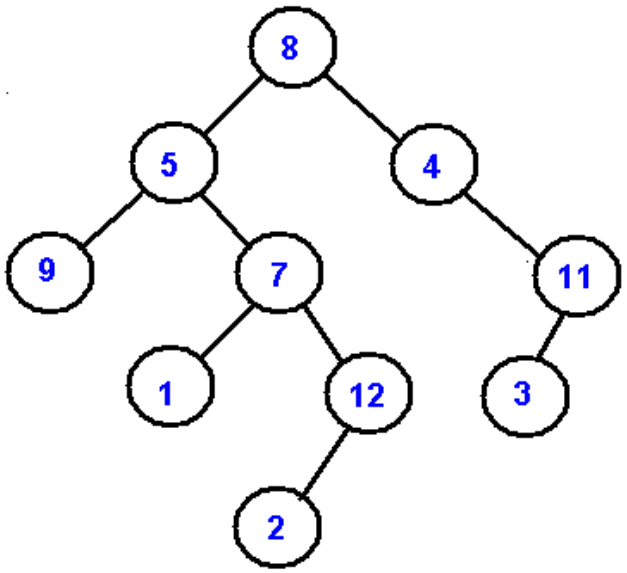
Tutorial 9

Exercise 1

Tree in Linked List

```
class Node:
    def __init__(self, data):
      self.left = None
      self.right = None
      self.data = data

    def traverseInorder(self, root):
        #traverse function will print all the node in the tree.
        if root is not None:
            self.traverseInorder(root.left)
            print(root.data)
            self.traverseInorder(root.right)

    def traversePreorder(self, root):
        #traverse function will print all the node in the tree.
        if root is not None:
            print(root.data)
            self.traversePreorder(root.left)
            self.traversePreorder(root.right)

    def traversePostorder(self, root):
        #traverse function will print all the node in the tree.
        if root is not None:
            self.traversePostorder(root.left)
            self.traversePostorder(root.right)
            print(root.data)

root = Node(1)
root.left = Node(2)
root.right = Node(3)
root.left.left = Node(4)
root.left.right = Node(5)

print("Inorder traversal of binary tree is ")
root.traverseInorder(root)
print("Preorder traversal of binary tree is ")
root.traversePreorder(root)
print("Postorder traversal of binary tree is ")
root.traversePostorder(root)
```

Given the above class,



a) Show the sequence of elements in preorder traversal.
b) Show the sequence of elements in in-order traversal.
c) Show the sequence of elements in postorder traversal.
d) **revise given code that the following tree is implemented**

Exercise 3

For the expression (1 + 2) * (8 – 3) / (5 + 7)

(a) Draw the expression tree

(b) Give prefix and postfix forms of the expression

(c) Which tree traversal method gives prefix form of expression?

(d) Which tree traversal method gives postfix form of expression?

(e) What are the differences between an expression tree and a tree?

(f) Figure 1 shows the first three operations of postfix evaluation using stack data structure for scanning the postfix expression obtained in part (b) of this question.

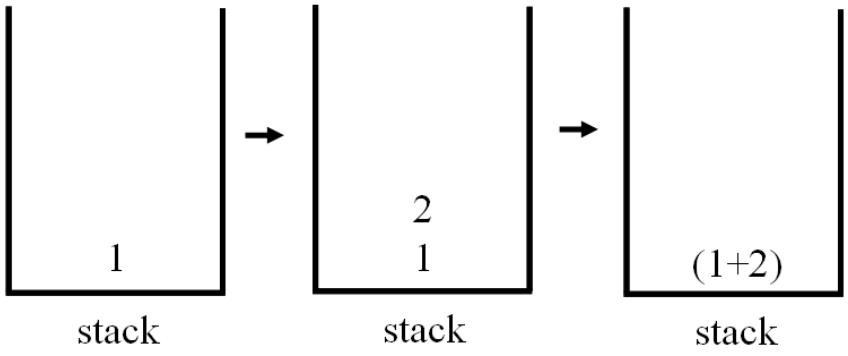


Figure 1

Show the rest operations of postfix evaluation using stack for the postfix expression obtained in part (b) of this question.